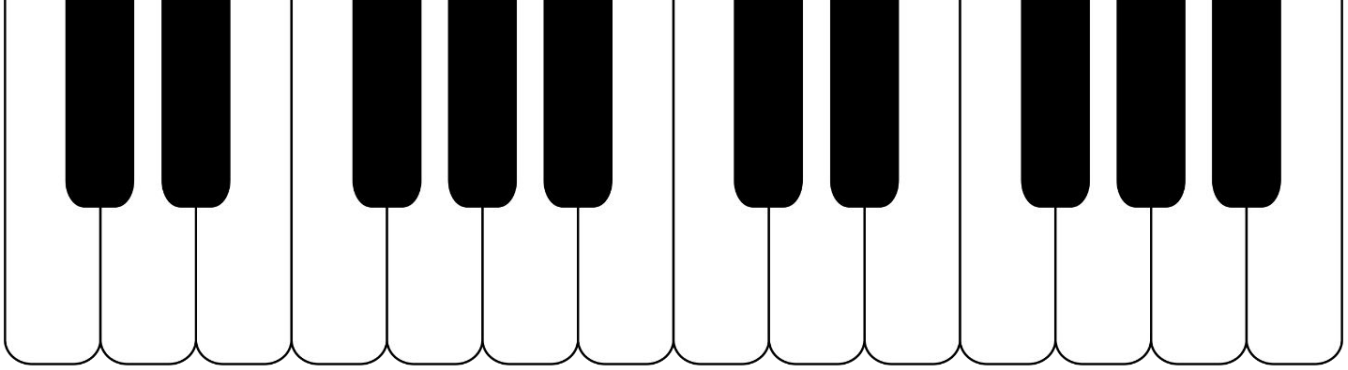
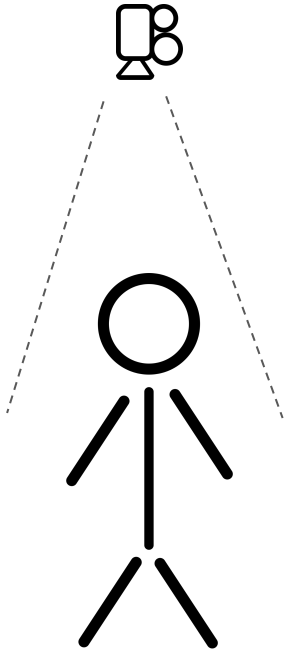


Team D2: Keynetic

Lance Yarlott, Katherine Dettmer, Sun A Cho

The Device

- Mechanically actuated keyboard that is managed by a microcontroller
- Provides a way to play the piano without needing the physical ability to press keys
- Limited to simple notes and chords
- Playing range = two octaves on piano keyboard, only white keys
- Currently, there are no widespread solutions for playing the piano without pressing keys or generating sound directly from a computer



Requirement #1: Hardware/Mechanism

- Design an external electrical system to play the keyboard
  - Designing an electrical system that can support multiple solenoids
  - Designing a program that could support turning on/ off these electrical components (i.e. actuators)

- Build a successfully, functioning external actuator system for the white keys
  - Using 14 solenoids
  - Similar to self-playing pianos in the market
  - Easy to use + install on the existing keyboard/ piano

Requirement #2: Hand/Feature Detection

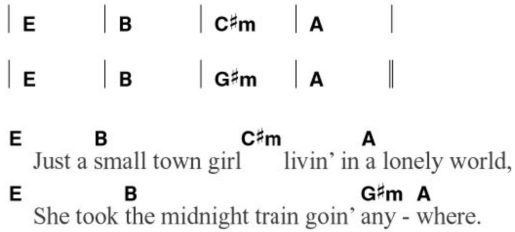
- Detect hand or other marked feature position > 90% of the time
  - Design a system that uses hand location on the screen to determine notes
  - Recognize hand/feature positions when they are > 4 ft away and < 7 ft away

- Detect pattern a hand makes > 90% of the time
  - Design grid system that records where the hand has been and sees if at any point the sequential pattern matches the patterns we have mapped

- **Goal:** Reduce latency as much as possible
  - Minimum Requirement: Response time (from time user makes motion to when the key is pressed) < 1 second
  - Long response time would take away from the feeling of actually playing an instrument
  - Goal requirement: As close to 0s as possible

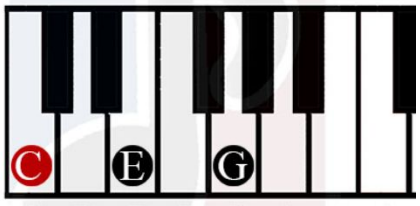
Requirement #3: Music Composition Generation

- Generate chord progressions in the key of C major
  - Estimate what progression is being played based on past notes
  - Progressions are cyclical over time



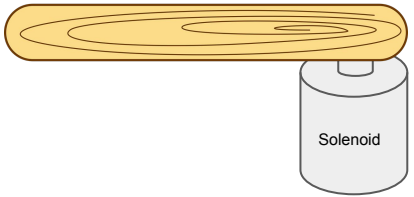
- On the downbeat, notes should generally fall on a chord tone
  - When generating progressions, try to make sure notes land on chord tones

- Should also be able to generate melodies over chord progressions
  - Should be relatively simple

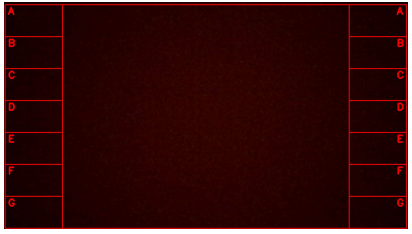


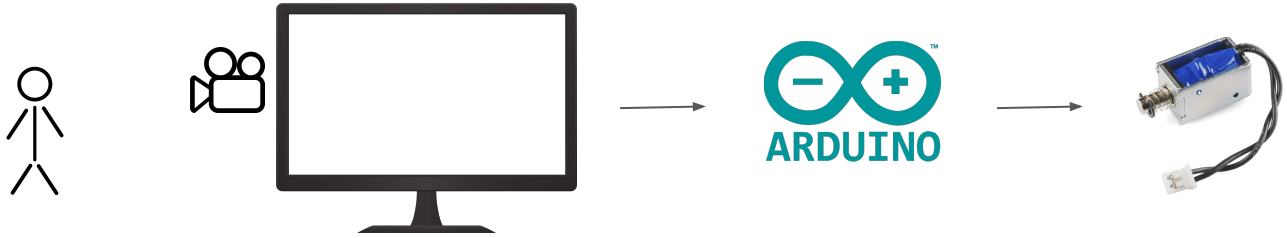
Solution Approach

- Power only the needed number of actuators
  - Avoids the use of unnecessary power



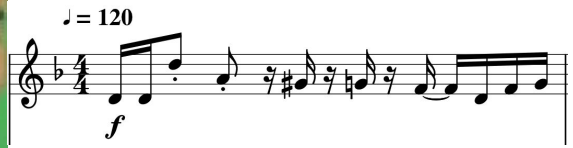
- Use OpenCV to track hand position in screen space
  - Divide the screen into grids, then detect what grid the hand is in
  - Then, we can track the hand for some period, record the pattern and figure out what gesture they made based on the patterns we have stored
  - Switch between single-note mode and pattern mode with reaching hand into certain area of screen (like a button)





Solution Approach (contd.)

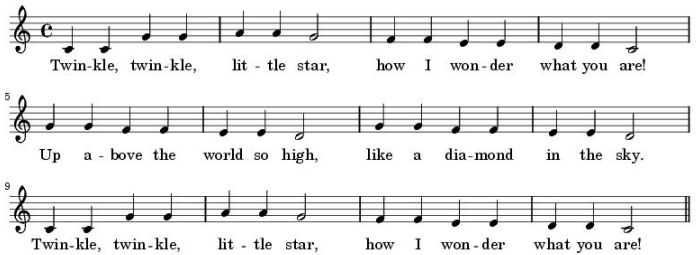
- Music Generation
  - Have users directly control notes they play
    - Also supports sheet music
  - Support for subdivisions and rests
  - For two handed melody control, use Bayesian Updating, estimate chord progression
    - “Lookahead” for expected notes in case of inconsistent playing



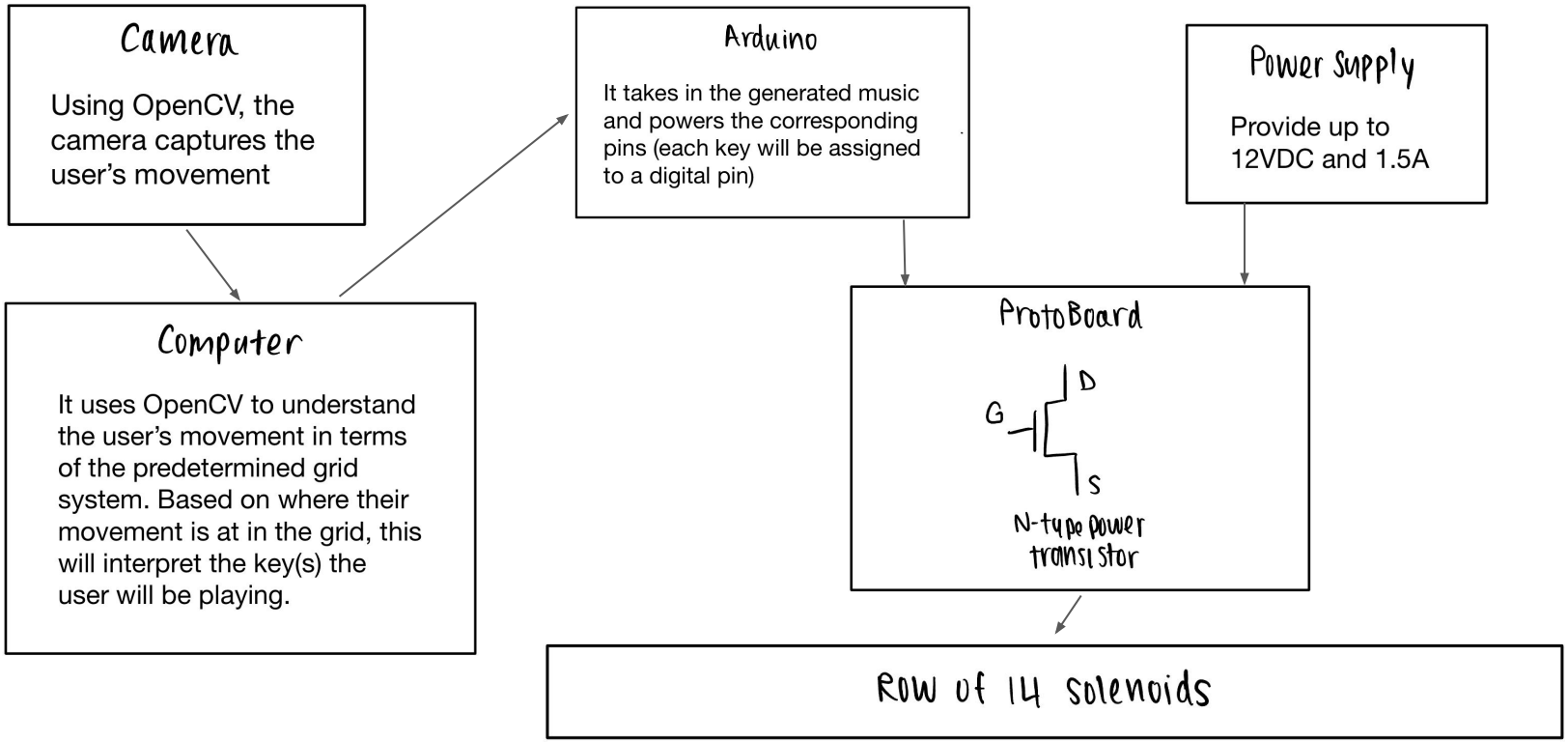
- Opportunities to experience the joy of playing music
  - Target audience: people who do not have the traditional physical ability to play music

Testing, Verification and Metrics

- Hardware: playing Twinkle, Twinkle Little Star correctly, with chords
- Computer Vision/ SW: measure accuracy of recognizing hands and their positions on the screen
  - Measure accuracy of recognizing patterns in the grid
- Measure response time from when user makes motion to when the key is played (goal: < 1 sec)
- Music Generation: Correctly pass notes to hardware at correct tempo (at least 60 bpm)

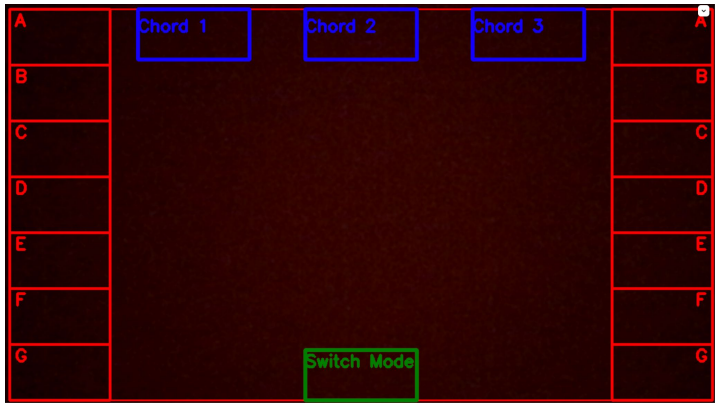


Block Diagram

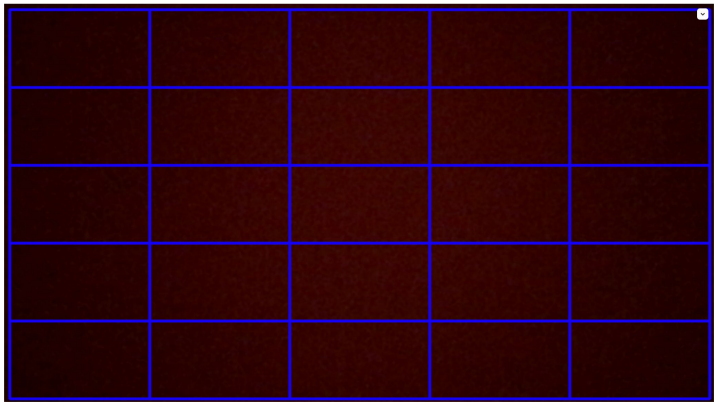


Implementation Plan

- OpenCV: hand/motion detection
  - Creating color detection small enough to hit accurate notes
  - Designing key system for normal play mode, grid system for generative
  - Designing and implementing generation of position to notes, switching between note mode and generative mode



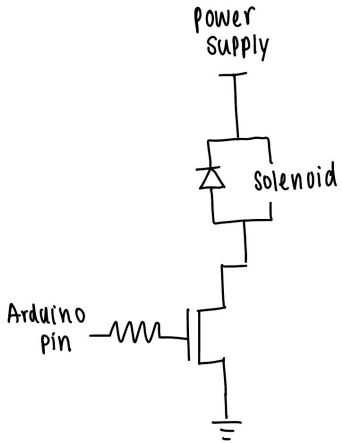
Normal



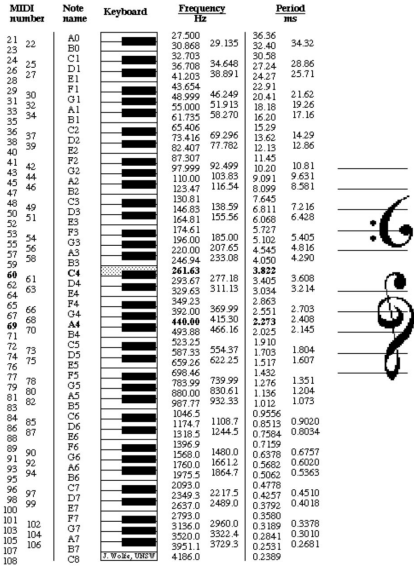
Generative

Implementation Plan (Cont.)

- Actuator/Solenoid: build the actuator system to play the piano



- Music: chord estimation & melody representation
  - Encode notes as MIDI numbers for consistency
  - Send chord signals simultaneously
  - Notes represented as start time and duration



Motional Keyboard

The Keyboarders

Tom Sullivan

Project Start Date: 1/30/23

Scrolling Increment: 0

Legend: On track | Low risk | Med risk | High risk | Unassigned

Milestone description	Category	Assigned to	Progress	Start	Days
Keyboard Hardware					
MIDI Keyboard Procurement	Milestone	Name	15%	2/13/23	1
Frame Creation	Low Risk	Suna	0%	2/14/23	7
Actuator Selection	Med Risk	Suna	5%	2/6/23	7
Actuator Integration	High Risk	Suna	0%	2/21/23	7
Software Integration	Med Risk	Suna	0%	2/28/23	7
CV					
Detection Method Selection	Low Risk	Katherine	5%	1/30/23	7
Hand Detection	High Risk	Katherine	0%	2/6/23	7
Gesture Mapping	Med Risk	Katherine	0%	2/13/23	7
Multi-feature Detection	Low Risk	Katherine	0%	2/20/23	7
Music Software Integration	Goal	Katherine	0%	2/27/23	1
Music Synthesis					
Sound Generation Algorithm	On Track	Lance	5%	2/13/23	6
Note Sequencing (In-time Quarter Notes)	High Risk	Lance	0%	2/13/23	6
Pitch Selection	High Risk	Lance	0%	2/19/23	6
Chord Playing	Med Risk	Lance	0%	2/25/23	7
Advanced Music Synthesis					
Subdivions	Low Risk	Lance	0%	2/19/23	7
Syncopation	Low Risk	Lance	0%	2/26/23	7
Chords (Non-Triads)	Low Risk	Lance	0%	3/5/23	7
Verification and Validation					
Unit Testing	Med Risk	Lance	0%	3/6/23	5
Hardware Testing	Low Risk	Suna	0%	2/28/23	5
Software Testing	Low Risk	Katherine	0%	3/7/23	5
Music Synthesis Testing	Med Risk	Lance	0%	3/4/23	7
Integration Testing	High Risk	All	0%	3/11/23	7
Slack Creation	Low Risk	All	0%	2/5/23	1

To add more data, Insert new rows ABOVE this one